



ITS

Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

PENGENALAN EKSPRESI WAJAH UNTUK TUNANETRA MENGUNAKAN *DEEP LEARNING* PADA PERANGKAT PORTABEL

Hasby Fahrudin
NRP 07111540000112

Dosen Pembimbing
Dr.Ir. Hendra Kusuma, M.Eng.Sc.
Muhammad Attamimi, B.Eng, M.Eng., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - EE 184801

**Pengenalan Ekspresi Wajah untuk Tunanetra
Menggunakan *Deep Learning* pada Perangkat Portabel**

Hasby Fahrudin
NRP 07111540000112

Dosen Pembimbing
Dr.Ir. Hendra Kusuma, M.Eng.Sc.
Muhammad Attamimi, B.Eng, M.Eng., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



ITS

Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EE 184801

FACIAL EXPRESSIONS RECOGNITION FOR VISUALLY IMPAIRED PERSON USING DEEP LEARNING ON PORTABLE DEVICE

Hasby Fahrudin
NRP 07111540000112

Supervisors
Dr.Ir. Hendra Kusuma, M.Eng.Sc.
Muhammad Attamimi, B.Eng, M.Eng., Ph.D.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

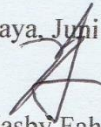
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi keseluruhan Tugas akhir saya dengan judul **“PENGENALAN EKSPRESI WAJAH UNTUK TUNANETRA MENGGUNAKAN *DEEP LEARNING* PADA PERANGKAT PORTABEL”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2019



Hasby Fahrudin
07111540000112

Halaman ini sengaja dikosongkan

PENGENALAN EKSPRESI WAJAH UNTUK TUNANETRA MENGGUNAKAN *DEEP LEARNING* PADA PERANGKAT PORTABEL

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Elektronika

Departemen Teknik Elektro

Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I



Dosen Pembimbing II



Dr.Ir. Hendra Kusuma, M.Eng.Sc.

Muhammad Attamimi, B.Eng, M.Eng.,

Ph.D.

NIP. 196409021989031003

NPP. 1985201711039



Halaman ini sengaja dikosongkan

ABSTRAK

Penyandang tunanetra menghadapi tantangan yang lebih dalam kehidupan sehari-hari seperti mencari jalan di lingkungan yang tidak dikenal, mendeteksi objek dan orang, dan pengenalan wajah dan ekspresi wajah. Salah satunya adalah interaksi tatap muka yang dimana banyak informasi dipertukarkan secara nonverbal melalui postur tubuh, gerakan, kedekatan interpersonal dan ekspresi wajah. Ekspresi wajah dipercayai mempunyai kaitan erat dalam emosi seseorang dan memberikan informasi yang lebih lengkap saat berkomunikasi. Penyandang tunanetra tidak bisa mendapatkan informasi tersebut sehingga adanya batasan dalam interaksi tatap muka.

Penelitian tentang pengenalan ekspresi wajah kepada penyandang tunanetra telah dilakukan. Akan tetapi, kendala yang selalu dihadapi adalah biaya pembuatan dan desain perangkat yang kurang praktis. Pada penelitian ini akan dibuat sebuah perangkat yang lebih praktis dan lebih murah dengan performa yang baik. Dengan begitu, akan lebih banyak penyandang tunanetra yang terbantu. Pada penelitian ini dibuat perangkat pembantu pengenalan ekspresi wajah untuk tunanetra yang akan diimplementasikan ke perangkat yang relatif kecil yaitu raspberry pi. Perangkat akan menyampaikan informasi ke pengguna melalui suara. *Deep learning* menjadi bagian utama dalam pengenalan ekspresi wajah. Model deep learning akan dibuat menggunakan gabungan dataset yang sudah ada dan juga memanfaatkan *pre-trained network*. Perangkat yang didesain pada penelitian ini mempunyai akurasi mencapai 80% pada ruangan *indoor* dan 75% pada ruangan *outdoor*.

Kata kunci: Rekognisi Ekspresi Wajah, Tunanetra, Teknologi Bantu, *Deep Learning*, Perangkat Portabel

Halaman ini sengaja dikosongkan

ABSTRACT

Persons with visual impairments have more hurdle than normal person in daily life such as finding roads in unknown environments, detecting object and people, facial recognition and expressions. One of it is face-to-face interaction that involved non-verbal information through body gesture, interpersonal connection, and facial expressions. Facial expression believed have a strong bond in human emotion and gives more complete informations when communicate. Visually impaired persons people cannot have that information so they have hinderance while communication in face-to-face interraction.

There are reasearch about delivering facial expressions to visually impaired persons. But, the problem always about production cost and uncomfortable design. In this research we will make less cost device and comfortable with good performance. In this study, system to recognize facial expressions for blind people was made which would be implemented on a relatively small device, raspberry pi. The device will convey information to the user via voice. Deep learning is a major part of the facial expressions recognition system. The deep learning model will be created using a combination of existing datasets and make use of pre-trained network. Devices designed in this study have an accuracy of up to 80% in indoor and 75% in outdoor spaces.

Keyword: *Facial Expression Recognition, Visually Impaired, Assistive Technology, Deep Learning, Portable Devices*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji Syukur penulis panjatkan kehadiran Allah SWT atas segala Rahmat, Karunia, dan Petunjuk yang telah dilimpahkan-Nya sehingga penulis mampu menyelesaikan tugas akhir dengan judul **“PENGENALAN EKSPRESI WAJAH UNTUK TUNANETRA MENGGUNAKAN *DEEP LEARNING* PADA PERANGKAT PORTABEL”**.

Tugas Akhir ini disusun sebagai salah satu persyaratan untuk menyelesaikan jenjang pendidikan S1 pada Bidang Studi Teknik Sistem Elektronika, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Atas selesainya penyusunan tugas akhir ini, penulis mengucapkan terima kasih kepada:

1. Ibu dan Bapak penulis atas doa dan cinta yang tak henti pada penulis dalam keadaan apapun. Semoga selalu diberi kesehatan dan kebahagiaan.
2. Bapak Hendra Kusuma dan Bapak Muhammad Attamimi selaku dosen pembimbing yang telah memberikan arahan, bimbingan dan perhatiannya selama proses penyelesaian tugas akhir ini.
3. Seluruh dosen dan karyawan Departemen Teknik Elektro ITS yang telah memberikan banyak ilmu dan menciptakan suasana belajar yang luar biasa.
4. Teman-teman asisten Laboratorium Mikroelektronika dan Sistem Tertanam yang memberi motivasi dan semangat selama pengerjaan tugas akhir.
5. Teman-teman seperjuangan e55 yang telah menemani dan memberikan dukungan selama masa kuliah sampai penyusunan tugas akhir ini.

Penulis berharap penelitian pada tugas akhir ini dapat bermanfaat bagi ilmu pengetahuan khususnya di bidang kecerdasan buatan.

Surabaya, Juni 2019

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT.....	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Tugas Akhir.....	2
1.4 Batasan Masalah.....	2
1.5 Metodologi	2
1.6 Sistematika Penulisan	4
1.7 Relevansi	4
BAB 2 TINJAUAN PUSTAKA.....	5
2.1 Teori Penunjang	5
2.1.1 <i>Artificial Neural Network</i>	5
2.1.2 Deep Learning	7
2.1.3 <i>Convolutional Neural Network</i>	8
2.1.4 Komputer Visi	12
2.1.5 TensorFlow	12
2.1.6 Keras	13
2.1.7 Raspberry Pi	14
2.1.8 OpenCV.....	15
2.1.9 Anaconda.....	15
2.1.10 Dataset Penunjang	17
2.2 Penelitian Sebelumnya	20

2.2.1	<i>A Face Recognition Application for People with Visual Impairments: Understanding Use Beyond the Lab [21]</i>	20
2.2.2	<i>A Smart Cane: Face Recognition System for Blind [22]</i>	21
2.2.3	<i>Conveying Facial Expressions to Blind and Visually Impaired Persons Through a Wearable Vibrotactile Device [23]</i>	21
BAB 3	METODOLOGI PENELITIAN	23
3.1	Diagram Blok Sistem	23
3.2	Akuisisi Gambar	24
3.3	<i>Image Preprocessing</i> dan Ekstraksi Fitur	24
3.4	Pengenalan Ekspresi Wajah dengan <i>Deep Learning</i>	25
3.4.1	Pembuatan Dataset	25
3.4.2	Melatih Model <i>Deep Learning</i>	26
3.5	Informasi Suara	28
3.6	Desain Perangkat	29
3.6.1	Rancangan Kemasan Perangkat	29
3.6.2	Skenario Pemakaian	30
3.7	Pengujian Perangkat	31
BAB 4	HASIL PENGUJIAN	33
4.1	Kecepatan Prediksi Model <i>Deep Learning</i>	33
4.2	<i>Hyperparameter Tuning</i>	34
4.2.1	Tuning Momentum	34
4.2.2	Tuning Learning Rate	36
4.2.3	Model Akhir	37
4.3	Pengujian Perangkat Pada Pengguna Dengan Pengelihat Normal	38
4.3.1	Pengujian Di Dalam Ruangan	38
4.3.2	Pengujian Di Luar Ruangan	40
4.4	Pengujian Perangkat Pada Pengguna Tunanetra	42
4.4.1	Pengujian Di Dalam Ruangan	42

4.4.2 Pengujian Di Luar Ruangan	44
BAB 5 PENUTUP.....	47
5.1 Kesimpulan.....	47
5.2 Saran.....	47
DAFTAR PUSTAKA	49
LAMPIRAN A.....	52
LAMPIRAN B	63
BIODATA PENULIS	67

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2 .1 Deskripsi fungsional dari struktur neuron [1].	5
Gambar 2.2 Contoh arsitektur neural network [1].	6
Gambar 2.3 Contoh fungsi aktivasi: (a)Sigmoid (b)Tanh (c)ReLu [1].	6
Gambar 2.4 Diagram proses backpropagation.	7
Gambar 2.5 Performa dari deep learning (DNN) dan algoritma pembelajaran mesin lainnya (GBDT, LR, SVM) dengan jumlah data latih yang berbeda [2].	8
Gambar 2.6 Arsitektur CNN sederhana [3].	9
Gambar 2.7 Arsitektur Xception [5].	10
Gambar 2.8 Blok Diagram MobileNetV2 [6].	11
Gambar 2.9 Arsitektur VGG16 [7].	11
Gambar 2.10 Logo TensorFlow [10].	13
Gambar 2.11 Logo Keras [12].	14
Gambar 2.12 Raspberry Pi 3 Model B [13].	14
Gambar 2.13 Logo OpenCV.	15
Gambar 2.14 Logo Anaconda [15].	16
Gambar 2.15 Logo Spyder [15].	16
Gambar 2.16 Tampilan Anaconda Navigator.	16
Gambar 2.17 Contoh Workspace pada Spyder.	17
Gambar 2.18 Contoh gambar pada dataset CK+ [16].	18
Gambar 2.19 Contoh anotasi pada database AffectNet [17].	18
Gambar 2.20 Contoh gambar pada database.	19
Gambar 2.21 Penggunaan accessibility bot [21].	20
Gambar 2.22 Implementasi perangkat [22].	21
Gambar 2.23 Sistem substitusi sensorik yang dipakai dalam penelitian [23].	22
Gambar 3.1 Diagram Blok Sistem.	23
Gambar 3.2 Penempatan webcam pada topi.	24
Gambar 3.3 Proses <i>image preprocessing</i> dan ekstraksi fitur.	25
Gambar 3.4 Struktur Penyimpanan Dataset.	26
Gambar 3.5 Ilustrasi dari Modifikasi Arsitektur Deep Learning.	27
Gambar 3.6 Spesifikasi dari MobileNetV2 yang Dimodifikasi pada Keras.	28
Gambar 3.7 Output dari Pelatihan Model Deep Learning.	28
Gambar 3.8 Program Untuk Mengatur Parameter Learning.	28
Gambar 3.9 Rancangan perangkat tampak depan.	30
Gambar 3.10 Rancangan perangkat tampak atas.	30
Gambar 3.11 Ilustrasi skenario pemakaian perangkat.	31
Gambar 3.12 Contoh pemakaian perangkat.	31
Gambar 3.13 Contoh lokasi pengujian <i>outdoor</i> .	32
Gambar 3.14 Contoh lokasi pengujian <i>indoor</i> .	32
Gambar 4.1 Riwayat learning dengan nilai momentum 0,5.	34
Gambar 4.2 Riwayat learning dengan nilai momentum 0,7.	35

Gambar 4.3 Riwayat learning dengan nilai momentum 0,9.....	35
Gambar 4.4 Riwayat learning dengan nilai learning rate 0,001.	36
Gambar 4.5 Riwayat learning dengan nilai learning rate 0,0001.	36
Gambar 4.6 Riwayat learning dengan nilai learning rate 0,00001.	37
Gambar 4.7 Riwayat <i>learning</i> setelah melakukan <i>hyperparameter tuning</i>	38
Gambar 4.8 Pengujian sistem rekognisi ekspresi wajah dalam ruangan.....	39
Gambar 4.9 Diagram persentase keberhasilan prediksi pada pengujian dalam ruangan.	39
Gambar 4.10 Confusion matrix dari hasil pengujian sistem rekognisi ekspresi wajah dalam ruangan.	39
Gambar 4.11 Pengujian sistem rekognisi ekspresi wajah luar ruangan.	40
Gambar 4.12 Diagram persentase keberhasilan prediksi pada pengujian luar ruangan.	41
Gambar 4.13 Confusion matrix dari hasil pengujian sistem rekognisi ekspresi wajah luar ruangan.....	41
Gambar 4.14 Pengujian perangkat pada penyandang tunanetra di ruangan tertutup.....	42
Gambar 4.15 Diagram persentase keberhasilan prediksi pengguna tunanetra pada ruangan tertutup.	43
Gambar 4.16 Confusion matrix dari hasil pengujian perangkat ruangan tertutup.	43
Gambar 4.17 Pengujian perangkat pada penyandang tunanetra di ruangan terbuka.	44
Gambar 4.18 Diagram persentase keberhasilan prediksi pengguna tunanetra pada ruangan terbuka.	44
Gambar 4.19 Confusion matrix dari hasil pengujian perangkat ruangan tertutup.	45

DAFTAR TABEL

Tabel 3.1. Informasi Audio	29
Tabel 4.1 Hasil pengujian pada komputer uji.	33
Tabel 4.2 Hasil pengujian pada Raspberry Pi 3.	33

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Pada bab ini membahas mengenai hal-hal yang mendasari penelitian dengan penjelasan mendetail. Hal tersebut meliputi latar belakang, perumusan masalah, tujuan penulisan, batasan masalah, metodologi penelitian, sistematika penulisan, dan relevansi.

1.1. Latar Belakang

Penyandang tunanetra menghadapi tantangan yang lebih kompleks dalam kehidupan sehari-hari seperti mencari jalan di lingkungan yang tidak dikenal, mendeteksi objek, pengenalan wajah dan ekspresi wajah. Pada interaksi tatap muka terjadi pertukaran informasi secara nonverbal melalui postur tubuh, gerakan, kedekatan interpersonal dan ekspresi wajah. Ekspresi wajah dipercaya mempunyai kaitan erat dalam emosi seseorang dan memberikan informasi yang lebih lengkap saat berkomunikasi. Penyandang tunanetra tidak bisa mendapatkan informasi tersebut sehingga terjadi keterbatasan dalam berinteraksi tatap muka.

Tanpa adanya kemampuan untuk melihat, manusia dapat memproses informasi yang biasanya diperoleh melalui pengelihatian dengan menggunakan indera lain, seperti indera pendengaran atau peraba [1]. Untuk informasi seperti warna, tertulis, nonverbal, atau isyarat dapat diperoleh melalui pendengaran atau *haptic*. Contoh yang paling umum adalah braile, yang banyak digunakan untuk menyampaikan informasi tertulis. Pernah diadakan penelitian tentang menyampaikan informasi visual untuk penyandang tunanetra menggunakan sabuk *haptic* [2]. Namun, dalam segi desain perangkat yang dibuat kurang praktis dikarenakan bentuknya yang besar dan juga harganya yang cukup mahal.

Faktor tersebut menyebabkan perangkat kurang dapat digunakan secara luas. Oleh karena itu, diperlukan alat yang lebih praktis dengan harga murah sehingga perangkat tersebut dapat mencakup lebih banyak pengguna. Dalam penelitian ini yang akan diperhatikan adalah biaya, ukuran, dan performa perangkat dengan mempertimbangkan *trade off* dari variabel tersebut.

1.2. Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Apa metode yang digunakan untuk mengklasifikasi emosi pada wajah?
2. Bagaimana cara menginformasikan hasil klasifikasi emosi kepada penyandang tunanetra?
3. Bagaimana desain perangkat yang murah dan praktis sehingga dapat digunakan penyandang tuna netra dengan baik?

1.3. Tujuan Tugas Akhir

Penelitian tugas akhir ini bertujuan untuk:

1. Untuk mendapatkan metode klasifikasi wajah yang efisien dan dapat diimplementasikan ke perangkat portabel.
2. Mengetahui antarmuka yang nyaman dan praktis digunakan oleh penyandang tunanetra.
3. Untuk mendapatkan desain perangkat yang murah dan portabel dengan performa baik.

1.4. Batasan Masalah

1. Desain perangkat hanya memperhatikan sisi fungsional.
2. Emosi wajah yang direkognisi adalah marah, bahagia, jijik, takut, terkejut, sedih, dan netral.
3. Performa perangkat diuji pada ruangan terbuka dan tertutup.

1.5. Metodologi

Penelitian ini akan dilakukan dalam beberapa langkah yaitu sebagai berikut:

1. Studi Literatur
Studi literatur akan berisi pengumpulan serta pengkajian teori, data dan penelitian yang dianggap relevan dan terpercaya untuk mendukung keabsahan tugas akhir ini. Tema literatur yang akan dipelajari berkaitan dengan implementasi deep learning, komputer visi, dan efek pengenalan ekspresi wajah pada tunanetra. Literatur yang

digunakan harus bersumber dari paper, jurnal, buku, maupun artikel yang berasal dari sumber terpercaya.

2. Perancangan dan Pembuatan Perangkat Keras
Sistem yang telah dirancang sebelumnya akan diimplementasikan ke perangkat yang lebih portable dan nyaman digunakan. Acuan penelitian dalam tahap ini adalah pengurangan biaya produksi tanpa mengurangi performa sistem secara signifikan dan kenyamanan dalam penggunaan.
3. Perancangan dan Pembuatan Perangkat Lunak
Setelah mempelajari teori, akan dirancang sistem yang sesuai dengan aplikasinya. Sistem akan dibagi menjadi 2 langkah yaitu lokalisasi wajah dan klasifikasi emosi. Pada bagian ini akan dilakukan uji perbandingan antara beberapa algoritma yang ada untuk menentukan algoritma yang cocok untuk diaplikasikan ke perangkat.
4. Pengujian Alat
Merupakan tahap pengujian keseluruhan sistem, apabila masih terdapat kesalahan pada sistem sehingga sistem tidak dapat berjalan dengan baik dan perlu revisi pada desain rangkaian atau perangkat lunaknya. Dilanjutkan dengan pengujian sistem secara keseluruhan untuk menghasilkan data final, yang merupakan tujuan dari sistem ini.
5. Analisis Data dan Evaluasi
Pada tahap ini, akan dilakukan Analisa terhadap data yang didapatkan. Sehingga didapatkan karakteristik dari perangkat yang telah dirancang dan dibuat. Analisa dilakukan performa sistem, apakah sudah memenuhi kriteria untuk digunakan. Lalu dilakukan Analisa terhadap data yang dihasilkan oleh keseluruhan sistem, apakah sudah dianggap memuaskan atau tidak. Lalu dilakukan evaluasi serta revisi desain apabila diperlukan.
6. Penyusunan Laporan
Penyusunan laporan akan dilakukan seiring dengan tahap-tahap lainnya. Isinya berkaitan dengan tugas akhir yang dikerjakan, meliputi pendahuluan, studi literatur, perancangan dan pembuatan sistem, pengujian dan Analisa serta penutup.

1.6. Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir ini terbagi atas lima bagian dan masing-masing bab terurai sebagai berikut:

BAB 1 Pendahuluan

Bab ini berisi penjelasan tentang latar belakang, permasalahan, tujuan, metodologi, sistematika penulisan, dan relevansi Tugas Akhir ini.

BAB 2 Tinjauan Pustaka

Pada bab ini berisi mengenai teori yang mendasari penyusunan laporan tugas akhir secara umum khususnya yang berhubungan komponen dan algoritma yang akan digunakan.

BAB 3 Perancangan Sistem

Bab ini berisi mengenai perancangan sistem rekognisi emosi secara visual dan juga desain perangkat keras.

BAB 4 Pengujian dan Analisis Data

Perangkat akan diuji performanya pada pengguna dengan pengelihat normal dan tunanetra.

BAB 5 Kesimpulan dan Saran

Bab ini berisi kesimpulan dari analisis yang dilakukan dan berisi tentang saran untuk pengembangan selanjutnya.

1.7. Relevansi

Penelitian ini diharapkan dapat membantu penyandang tunanetra dalam bersosialisasi dengan perangkat yang portabel. Dengan memanfaatkan teknologi terbaru, perangkat ini dapat memiliki performa yang baik meskipun dengan harga yang relatif lebih murah dan juga nyaman digunakan. Menjadi referensi bagi mahasiswa dalam penelitian di bidang serupa.

BAB 2

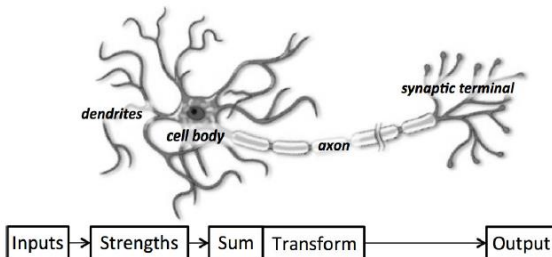
TINJAUAN PUSTAKA

2.1 Teori Penunjang

Pada subbab ini akan dibahas teori penunjang yang berkaitan dengan pembuatan perangkat keras dan lunak pada penelitian ini.

2.1.1 *Artificial Neural Network*

Artificial Neural Network adalah paradigma pemrosesan informasi yang biasanya digunakan untuk mempelajari tingkah laku suatu sistem yang kompleks pada simulasi komputer. Metode ini diilhami oleh otak manusia dalam memproses informasi. Tujuan dari *Artificial Neural Network* adalah memecahkan suatu masalah spesifik dengan alur yang sama dengan otak manusia.

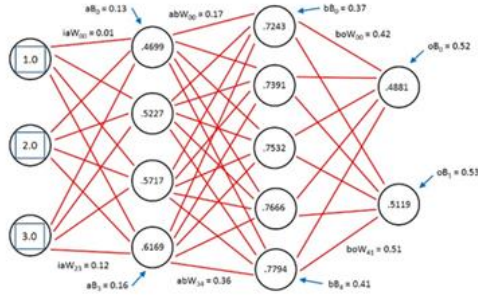


Gambar 2 .1 Deskripsi fungsional dari struktur neuron [1].

Metode ini mengimplementasikan kerja satu neuron dengan neuron yang lain pada program komputer. Model matematika dari satu unit neuron adalah :

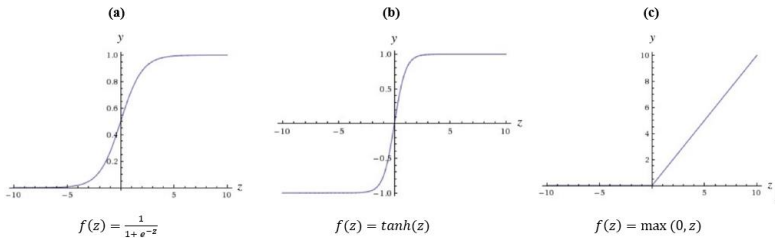
$$a = f(wp + b) \quad (2.1)$$

Input skalar p ditransmisikan melalui koneksi yang kekuatannya dikalikan oleh skalar *weight* (w). Hasil perkalian tersebut ditambahkan dengan skalar bias (b). Proses tersebut menghasilkan luaran skalar (a) yang nilainya akan diteruskan ke neuron selanjutnya.



Gambar 2.2 Contoh arsitektur neural network [1].

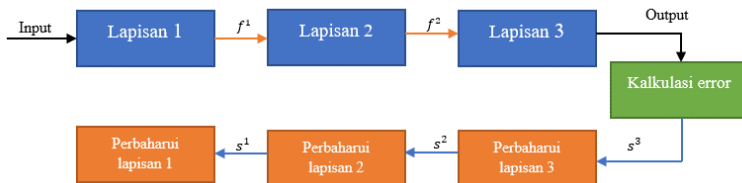
Untuk mempelajari hubungan neuron yang kompleks, diperlukan fungsi yang non-linear. Fungsi tersebut adalah fungsi aktivasi. Setiap lapisan dapat memiliki fungsi aktivasi yang berbeda sehingga setiap lapisan memiliki perilaku yang berbeda pula.



Gambar 2.3 Contoh fungsi aktivasi: (a)Sigmoid (b)Tanh (c)ReLU [1].

Proses otak manusia dalam mempelajari sesuatu dapat dimodelkan oleh *neural network* menggunakan *backpropagation*. *Backpropagation* adalah propagasi balik error prediksi model sebagai nilai acuan untuk memperbaharui *weight* dan *bias* pada setiap lapisan pada arsitektur *neural network*. Error tersebut didapat dengan membandingkan luaran model dengan target luaran yang sudah ditentukan sebelumnya. Nilai tersebut akan digunakan untuk mengkoreksi neuron dari lapisan terakhir ke

lapisan pertama. Perbaharuan tersebut untuk memodelkan proses belajar pada otak manusia.



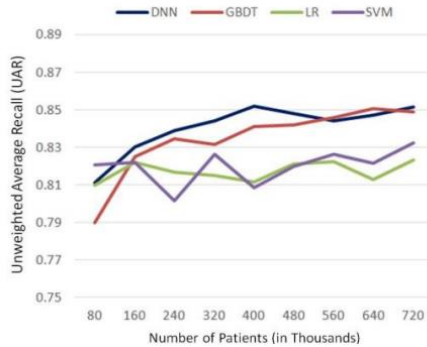
Gambar 2.4 Diagram proses backpropagation.

Untuk memperbarui *weight* dan *bias* pada setiap lapisan, diperlukan nilai sensitivitas propagasi (s^M) yang didapat dari lapisan setelahnya. Oleh karena itu, nilai terbaru dari suatu lapisan neuron akan mempengaruhi nilai terbaru dari lapisan sebelumnya. Lapisan terakhir akan mendapat nilai sensitivitas propagasinya dari nilai error.

2.1.2 Deep Learning

Deep learning adalah subbidang pembelajaran mesin yang berkaitan dengan algoritma yang terinspirasi oleh struktur dan fungsi otak yang disebut *artificial neural network*. Pada *deep learning*, model komputer dapat belajar untuk melakukan klasifikasi secara langsung dari gambar, tulisan, maupun suara. Performa *deep learning* terbukti sangat baik, bahkan dapat melebihi kemampuan manusia. Model dilatih menggunakan data yang sangat banyak dan arsitektur jaringan saraf yang memiliki banyak lapisan.

Inti dari *deep learning* adalah adanya komputer yang cukup cepat dan banyaknya data untuk melatih *artificial neural network* yang cukup besar secara efektif. Salah satu kekuatan *deep learning* adalah skalabilitas. Ketika kita membangun *artificial neural network* yang cukup besar dan melatihnya dengan semakin banyak data, performanya akan terus meningkat. Hal ini berbeda denganya algoritma pembelajaran mesin lainnya yang akan mencapai puncak performa pada posisi tertentu.

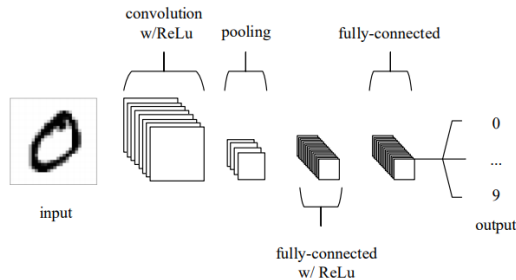


Gambar 2.5 Performa dari deep learning (DNN) dan algoritma pembelajaran mesin lainnya (GBDT, LR, SVM) dengan jumlah data latih yang berbeda [2].

Terdapat beberapa tipe artificial neural network pada penggunaan deep learning contohnya adalah *convolutional neural network* (CNN), *recursive neural network*, *recurrent neural network*, *long short-term memory* (LSTM), dan lain-lain. Pada penelitian ini akan diteliti lebih lanjut karena telah digunakan secara umum untuk pengenalan gambar.

2.1.3 Convolutional Neural Network

Indra pengelihat manusia sangatlah kompleks. Tidak hanya manusia dapat mengidentifikasi suatu objek, manusia dapat mengetahui kedalaman objek, mengetahui kontur suatu, dan juga memisahkan objek terhadap latar belakangnya. Hal ini dapat terjadi dikarenakan sambungan neuron manusia yang sangat kompleks. Untuk meniru perilaku neuron tersebut, kita tidak bisa memakai arsitektur *neural network* biasanya. Oleh karena itu, ditemukannya arsitektur yang dapat menangani gambar sebagai masukan yaitu CNN (*Convolutional Neural Network*).



Gambar 2.6 Arsitektur CNN sederhana [3].

CNN terdiri dari 3 lapisan yaitu *convolutional*, *pooling*, dan *fully-connected*. Fungsi dasar dari CNN bisa dibagi menjadi 4 proses [3]:

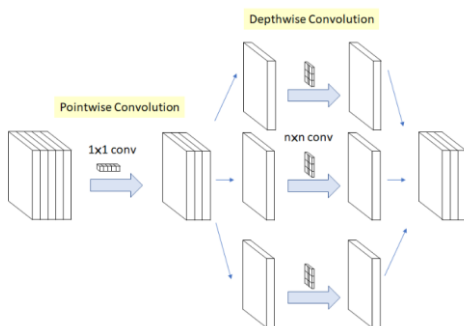
1. Gambar masuk pada *input layer* yang besarnya sama dengan jumlah pixel setiap saluran warna (RGB).
2. *Convolutional layer* akan menentukan output dari neuron yang akan dihubungkan ke bagian input local melalui perhitungan scalar antara *weights* dan bagian yang dihubungkan ke input. *Rectified Linear Unit* (ReLU) diterapkan pada output dari *layer* sebelumnya.
3. *Pooling layer* melakukan *downsampling* luaran *layer* sebelumnya sehingga mengurangi dimensi input dan mengurangi jumlah parameter dalam proses aktivasi tersebut.
4. *Fully-connected layer* arsitekturnya sama seperti *neural network* standar yang luarannya adalah nilai dari masing masing kelas. Lapisan ini diletakkan dibagian akhir dari arsitektur CNN, oleh karena itu dimensi masukannya lebih sederhana.

Dengan melewati proses ini, CNN dapat mengubah masukan gambar yang dimensinya kompleks menjadi lebih sederhana karena teknik *convolutional* dan *downsampling* untuk mengklasifikasi objek.

2.1.3.1 Xception

Xception adalah arsitektur CNN yang berbasis pemisahan *convolutional layer* secara mendalam. Hal ini menyebabkan pemetaan

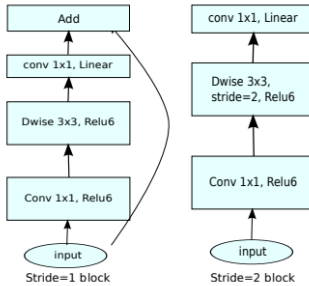
dari korelasi *cross-channels* dan korelasi spasial pada peta fitur CNN dapat dipisahkan. Arsitektur Xception adalah perbaruan dari arsitektur Inception. Arsitektur Xception mempunyai 36 *convolutional layer* yang membentuk basis ekstraksi fitur dari jaringan. 36 *convolutional layer* tersebut dibentuk menjadi 14 modul, yang dimana mempunyai koneksi residual linear di sekitarnya, kecuali modul pertama dan terakhir. Xception mempunyai akurasi yang lebih tinggi dari versi sebelumnya (Inception-v3) [4].



Gambar 2.7 Arsitektur Xception [5].

2.1.3.2 MobileNetV2

MobileNetV2 adalah arsitektur CNN yang didesain agar mampu dijalankan dalam perangkat mobil dengan tetap menjaga akurasi. Arsitektur ini berdasarkan pada struktur *inverted residual* dimana adanya jalan pintas diantara *bottleneck layer* yang tipis, lapisan ekspansi menengah menggunakan konvolusi mendalam yang ringan untuk memfilter fitur sebagai sumber non-linearitas.



Gambar 2.8 Blok Diagram MobileNetV2 [6].

Hal ini dilakukan untuk kekuatan representasional dari arsitektur. Oleh karena itu, arsitektur ini memiliki parameter yang lebih sedikit dengan menjaga performanya [6].

2.1.3.3 VGG16

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Gambar 2.9 Arsitektur VGG16 [7].

VGG16 adalah arsitektur CNN yang menambah kedalam arsitektur menggunakan convolutional filter kecil yang ternyata dalam proses pengujian performanya lebih baik dari konfigurasi yang sudah ada dengan menambah kedalam arsitektur menjadi 16-19 *weight layer*. VGG16 mempunyai lebih dari 100 juta parameter sehingga dibutuhkan komputer yang kuat untuk melatihnya [7].

2.1.4 Komputer Visi

Komputer visi adalah bidang ilmiah interdisipliner yang membahas bagaimana komputer dapat dibuat untuk memperoleh pemahaman tingkat tinggi dari gambar atau video digital. Dari perspektif teknik, ia berupaya mengotomatiskan tugas-tugas yang dapat dilakukan oleh sistem visual manusia.

Model yang digunakan pengelihatn mesin biasanya dikembangkan dalam fisik (radiometri, optic, dan desain sensor) dan dalam grafik komputer. Kedua bidang tersebut memodelkan bagaimana objek bergerak, bagaimana cahaya memantul dari permukaannya yang lalu dibiaskan melalui lensa kamera, dan diproyeksikan ke sebuah bidang gambar [8].

2.1.5 TensorFlow

TensorFlow adalah *open source software library* untuk komputasi berperforma tinggi. TensorFlow memiliki arsitektur yang sangat fleksibel sehingga dapat digunakan oleh banyak platform (GPU, CPU, TPU) dan dari *desktop* ke server hingga ke perangkat di ujung sistem. Pada awalnya dikembangkan oleh tim Google Brain dalam organisasi Google AI. TensorFlow berfokus pada *deep learning* dan *machine learning* dan juga inti komputasi numerik yang fleksible yang dapat digunakan pada domain saintifik lainnya. TensorFlow adalah *deep learning library* terpopuler saat ini. Perusahaan besar seperti google, nvidia, kakao, snapchat, dan twitter menggunakan tensorflow. Logo dari TensorFlow ditunjukkan pada Gambar 2.10 [9].



Gambar 2.10 Logo TensorFlow [10].

2.1.5.1 TensorFlow Lite

TensorFlow Lite (TFLite) adalah TensorFlow versi ringan untuk perangkat tertanam dan *mobile*. TFLite dapat menjalankan model machine learning pada perangkat berkomputasi rendah dengan latensi rendah dan ukuran biner yang kecil. TensorFlow Lite menggunakan beberapa teknik seperti mengoptimalkan kernel pada aplikasi ponsel, aktivasi *pre-fused*, dan kuantisasi kernel sehingga didapat model yang lebih kecil dan cepat. TensorFlow Lite memiliki model optimisasi. Metode yang digunakan adalah mengurangi jumlah parameter dengan *pruning*, mengurangi presisi representasi, dan memperbarui topologi model menjadi lebih efisien dengan mengurangi parameter atau mempercepat eksekusi sebagai contoh metode dekomposisi tensor dan distilasi [11].

2.1.6 Keras

Keras adalah API *neural-network* tingkat tinggi yang ditulis pada Bahasa pemrograman python dan mampu berjalan diatas TensorFlow, CNTK, atau Theano. Keras difokuskan untuk memfasilitasi eksperimen sederhana karena mempunyai API yang sangat sederhana dibandingkan API milik *back-end* dari Keras. Keras cocok untuk eksperimen yang cepat dan sederhana, mendukung *convolutional network* dan *recurrent network*, dan mampu berjalan pada CPU maupun GPU [12].



Gambar 2.11 Logo Keras [12].

2.1.7 Raspberry Pi

Raspberry Pi adalah komputer berukuran kecil, berharga murah, dan juga menggunakan keyboard dan mouse. Perangkat ini memungkinkan semua orang dari segala usia untuk menjelajahi komputasi, dan belajar bagaimana memprogram dalam bahasa seperti Scratch dan Python. Raspberry Pi mampu melakukan semua yang dapat dilakukan oleh komputer desktop, mulai dari menjelajah internet dan memutar video definisi tinggi, hingga membuat spreadsheet, pemrosesan kata, dan bermain game.



Gambar 2.12 Raspberry Pi 3 Model B [13].

Kecepatan prosesor berkisar dari 700 MHz hingga 1,4 GHz untuk Pi 3 Model B +. Papan memiliki satu hingga empat port USB. Untuk output video, HDMI dan video komposit didukung, dengan jack lengan tip tip standar 3,5 mm untuk output audio. Output tingkat lebih rendah disediakan oleh sejumlah pin GPIO, yang mendukung protokol umum seperti I²C [13].

2.1.8 OpenCV

Open Source Computer Vision Library (OpenCV) merupakan librari komputer visual dan machine learning. OpenCV dibuat untuk menyediakan infrastruktur umum untuk computer vision dan mempercepat produksi pada produk komersial. OpenCV mempermudah pengguna machine learning yang open source yang mempunyai lisensi BSD.

OpenCV mempunyai ribuan library yang telah dioptimalkan dan disederhanakan sehingga lebih mudah dipakai. OpenCV mempunyai library seperti pendeteksi objek, mendeteksi wajah, mendeteksi gerakan, memfilter warna, mengekstrak sudut, dan masih banyak lagi [14].



Gambar 2.13 Logo OpenCV.

2.1.9 Anaconda

Anaconda adalah salah satu platform data science yang terkenal pada pemrograman python dan R. Anaconda adalah open-source distribusi pemrograman python dan R untuk komputasi saintifik (data science, aplikasi machine learning, pemrosesan data skala besar, Analisa prediktif, dan lain-lain). Tujuan Anaconda adalah mempermudah manajemen paket paket dan distribusinya. Versi paket dikelola oleh sistem manajemen paket bernama Conda. Dalam Anaconda terdapat Anaconda Navigator yaitu desktop GUI yang termasuk dalam distribusi Anaconda yang memudahkan pengguna untuk membuka aplikasi dan mengelola paket Conda [24]. Salah satu aplikasi yang ada pada Anaconda Navigator adalah Spyder IDE. Spyder adalah environment saintifik yang menawarkan kombinasi unik dari fungsionalitas pengeditan, analisis, debugging, dan profil lanjutan dari alat pengembangan komprehensif

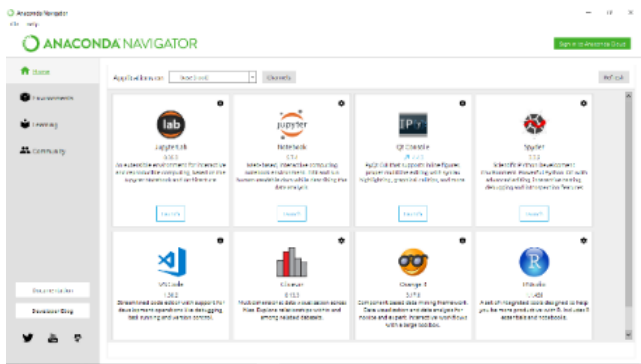
dengan eksplorasi data, eksekusi interaktif, inspeksi mendalam, dan kemampuan visualisasi yang indah dari paket ilmiah [15].



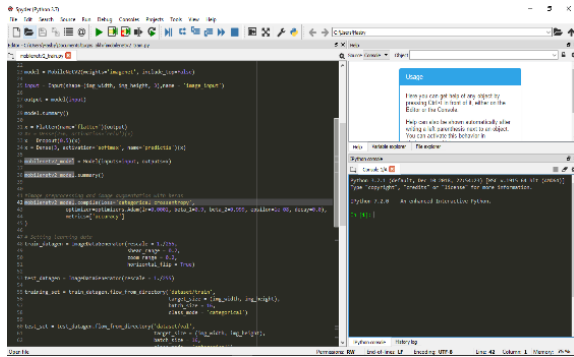
Gambar 2.14 Logo Anaconda [15].



Gambar 2.15 Logo Spyder [15].



Gambar 2.16 Tampilan Anaconda Navigator.



Gambar 2.17 Contoh Workspace pada Spyder.

2.1.10 Dataset Penunjang

Pada subbab ini akan dipaparkan dataset ekspresi wajah yang akan digunakan pada penelitian ini. Alasan digunakan beberapa dataset adalah untuk menambah variasi pada dataset latih dengan harapan meningkatkan akurasi model *deep learning* pada data nyata.

2.1.10.1 The Extended Cohn-Kanade Dataset (CK+) [16]

Cohn-Kanade dataset bertujuan untuk mempromosikan riset tentang deteksi ekspresi wajah individu secara otomatis. CK dataset telah menjadi salah satu dataset yang umum digunakan untuk pengembangan algoritma dan evaluasinya. Tetapi dataset ini masih ada 3 batasan yaitu: 1) label emosi tidak sesuai dengan permintaan dibandingkan dengan apa yang dilakukan, 2) Kurangnya metrik performa umum untuk mengevaluasi algoritma baru, dan 3) protokol standar untuk pembuatan dataset belum ada. Untuk mengatasi masalah tersebut dibuatlah ekstensi dari CK dataset yaitu CK+. Jumlah sekuen meningkat 22% dan jumlah subjek meningkat 27%. Target ekspresi untuk siap sekuen telah terkode FACS dan label emosi telah divalidasi dan direvisi.



Gambar 2.18 Contoh gambar pada dataset CK+ [16].

2.1.10.2 *AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild* [17]

AffectNet adalah database yang dibuat dengan cara *querying* mesin pencari yang berbeda (Google, Bing, dan Yahoo) menggunakan 1250 tag yang berhubungan dengan emosi pada enam bahasa yang berbeda (Inggris, Spanyol, Portugis, Jerman, Arab, dan Farsi). AffectNet mempunyai lebih dari 1 juta gambar dengan wajah. Dua belas orang ahli menganotasi masing masing 450.000 gambar dalam sisi kategori dan dimensi.



Gambar 2.19 Contoh anotasi pada database AffectNet [17].

Untuk menghitung kesamaan persepsi. 36.000 gambar dianotasi oleh dua orang. AffectNet sejauh ini menjadi database ekspresi wajah terbesar pada gambar statis yang mencakup model kategori maupun dimensi.

2.1.10.3 Japanese Female Facial Expressions (JAFFE) [18]

Database ini terdiri dari 213 gambar dari 5 ekspresi wajah (6 dasar ekspresi wajah + netral) dengan 10 wanita jepang sebagai modelnya. Setiap gambar dirating berdasar 6 emosi dari 60 subjek. JAFFE dirancang dan dibuat oleh Michael Lyons, Miyuki Kamachi, dan Jiro Gyoba. Foto diambil di departemen psikologi universitas Kyushu. Tiap ekspresor diambil gambarnya selagi melihat lembaran plastik semi-reflektif menuju kamera. Rambut diikat menjauhi wajah untuk menampilkan semua zona ekspresi pada wajah. Lampu tungsten diposisikan untuk mewujudkan untuk menambah penerangan di wajah. Sebuah kotak menutupi area antera kamera dan plastic untuk mengurangi *back-reflection*. Dataset Jaffe pernah digunakan sebagai data penelitian untuk membuktikan perbedaan budaya pada interpretasi ekspresi wajah. Pada penelitian tersebut disimpulkan bahwa adanya perbedaan interpretasi ekspresi wajah pada setiap orang yang memiliki identitas budaya berbeda [19]. Dengan menambahkan dataset JAFFE diharapkan mesin prediksi dapat beradaptasi pada masukan sistem yang variatif.

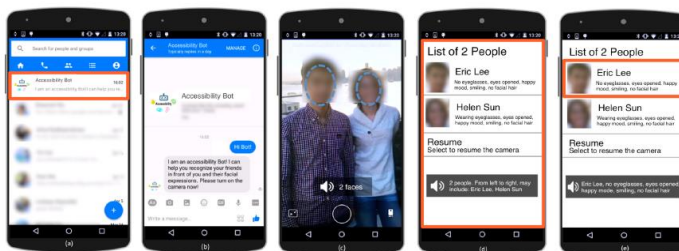


Gambar 2.20 Contoh gambar pada database.

2.2 Penelitian Sebelumnya

Pada subbab ini akan dipapar beberapa penelitian terdahulu yang mengilhami penelitian ini.

2.2.1 *A Face Recognition Application for People with Visual Impairments: Understanding Use Beyond the Lab* [21]

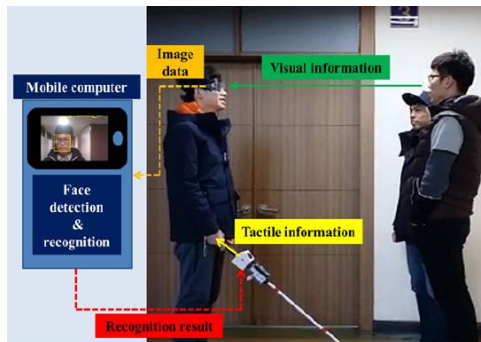


Gambar 2.21 Penggunaan accessibility bot [21].

Dalam penelitian ini, dibuat *Accessibility bot*, sebuah bot prototipe di Facebook Messenger yang memanfaatkan algoritma computer vision dan set foto yang ditandai dari teman pengguna di Facebook untuk membantu tunanetra mengenali teman-teman mereka. *Accessibility bot* memberikan informasi kepada pengguna tentang identitas teman di kamera dan ekspresi wajah mereka. Bot pada Facebook Messenger dapat dilatih dengan menggunakan foto pengguna yang ada di Facebook tanpa harus membuat dataset tersendiri. Bot bekerja dengan TalkBack, pembaca layar pada Android. Android dipilih karena memiliki pengguna yang banyak pada platform *smartphone*. Saat pengguna membuka aplikasi dan memulai berbicara dengan bot, bot akan membalas dengan pengenalan singkat, lalu menginstruksikan pengguna untuk menyalakan kamera untuk pengenalan wajah. Aplikasi ini terbatas hanya membantu penyandang tunanetra dalam bersosialisasi lewat media sosial online.

2.2.2 *A Smart Cane: Face Recognition System for Blind [22]*

Penelitian ini adalah desain prototipe alat bantu navigasi penyandang tunanetra berupa tongkat yang dapat mendeteksi wajah mengenalnya. Informasi disampaikan ke pengguna melalui perangkat getar yang ada pada tongkat. Pengguna memakai kacamata yang dipasang kamera untuk menangkap gambar. Gambar tersebut diproses di komputer menggunakan algoritma adaboost untuk mendapatkan hasil rekognisi wajah. Hasil dari wajah tersebut dikirimkan ke perangkat getar yang ada di tongkat menggunakan bluetooth. Perangkat getar akan menghasilkan pola getaran yang unik untuk setiap orang agar tunanetra dapat mengidentifikasi orang yang terdeteksi pada kamera. Kelemahan dari perangkat ini adalah bila label rekognisi wajah berjumlah banyak sehingga membutuhkan pola getar yang berbeda pada setiap label wajah. Hal ini dapat membingungkan pengguna karena harus mengingat pola getar pada setiap label wajah.

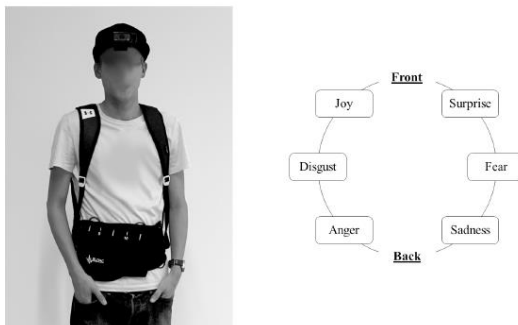


Gambar 2.22 Implementasi perangkat [22].

2.2.3 *Conveying Facial Expressions to Blind and Visually Impaired Persons Through a Wearable Vibrotactile Device [23]*

Pada penelitian ini dibuatlah perangkat pengenalan emosi wajah pada penyandang tunanetra menggunakan perangkat *vibrotactile* yang dapat dipakai. Emosi wajah yang dapat deteksi adalah marah, jijik, sedih, senang, terkejut, netral, dan takut. Perangkat ini menggunakan Microsoft Surface Pro 4 tablet sebagai pusat kontrol, Logitech HD Pro Webcam

C920 yang dipasang pada topi sebagai pengambil gambar pada jarak pandang pengguna. Sabuk *haptic* digunakan untuk membedakan emosi wajah satu dan yang lainnya. Gambar yang diambil dari kamera diterima oleh FaceReader 6™ yang keluarannya adalah prediksi emosi wajah. FaceReader 6™ adalah software yang menggunakan algoritma deteksi wajah *real-time* yang tangguh untuk mendeteksi wajah dari video dan *deep neural network* yang dapat mengklasifikasikan ekspresi wajah. Meskipun memiliki performa yang sangat baik, ukuran yang cukup besar menjadikan alat ini kurang praktis saat dipakai pada kegiatan sehari-hari.

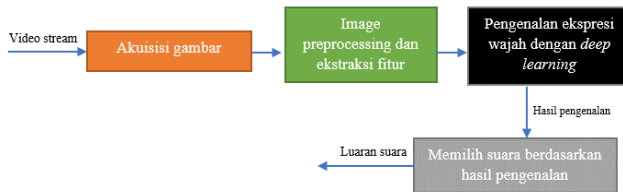


Gambar 2.23 Sistem substitusi sensorik yang dipakai dalam penelitian [23].

BAB 3

METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai perancangan sistem. Proses rekognisi ekspresi wajah diawali dengan pengambilan gambar wajah. Gambar yang telah diambil akan diteruskan ke unit pemproses. Sebelum dimasukan ke sistem prediksi, gambar akan diaugmentasi terlebih dahulu. Hasil prediksi akan disampaikan melalui informasi audio. Diagram blok sistem rekognisi ekspresi wajah yang telah didesain dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Blok Sistem.

3.1 Diagram Blok Sistem

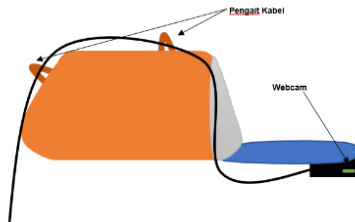
Berikut ini adalah penjelasan dari blok diagram dari perangkat rekognisi ekspresi wajah untuk penyandang tunanetra (lihat Gambar 3.1).

1. **Akuisisi Gambar**
Perangkat akan mengambil gambar melalui kamera yang telah terintegrasi pada perangkat.
2. **Image Preprocessing dan Ekstraksi Fitur**
Gambar yang diterima akan diaugmentasi sebelum dimasukkan pada sistem prediksi.
3. **Pengenalan Ekspresi Wajah dengan Deep Learning**
Gambar yang sudah diproses sebelumnya akan dimasukkan model *deep learning* untuk mengetahui label ekspresi wajah pada gambar. Luaran dari model akan berebentuk probabilitas tingkat keyakinan pada setiap label ekspresi wajah. Label yang memiliki tingkat keyakinan paling tinggi akan menjadi hasil prediksi.
4. **Infrormasi Suara**
Setiap label ekspresi wajah akan direpresentasikan pada file

audio yang memiliki komposisi nada berbeda. Informasi akan dikode menggunakan pola dari 3 macam nada (audio_A, audio_B, audio_C).

3.2 Akuisisi Gambar

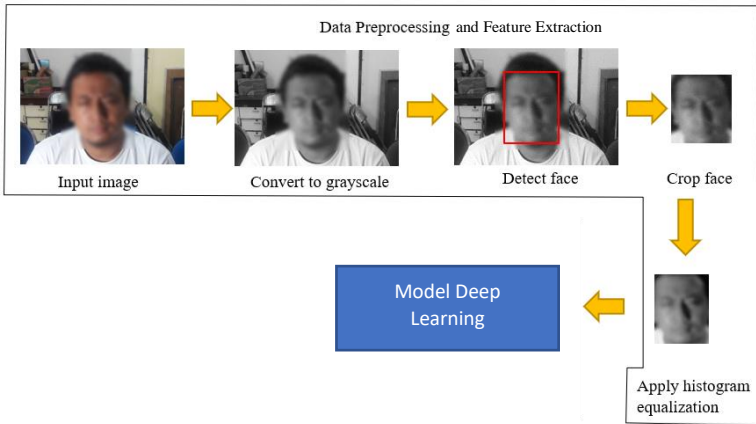
Masukan dari sistem adalah gambar yang ditangkap oleh webcam yang dimuatkan pada sebuah topi. Topi tersebut dipakai oleh pengguna perangkat. Kamera diletakkan sejajar dengan pandangan mata agar dapat menangkap wajah lawan bicara dengan jelas. Webcam yang digunakan adalah Logitech C170. Gambar akan dikirim via USB processing unit.



Gambar 3.2 Penempatan webcam pada topi.

3.3 *Image Preprocessing* dan Ekstraksi Fitur

Gambar yang diambil oleh webcam akan diproses pada python dengan bantuan OpenCV. OpenCV dengan bantuan Haar cascade dapat mendeteksi wajah manusia pada gambar tersebut. Bila terdeteksi, wajah dengan area terluas akan diambil dan diaplikasikan filter *histogram equalization*. Resolusinya diturunkan menjadi 200x200 pixel sehingga dapat dimasukkan pada model *deep learning*.



Gambar 3.3 Proses *image preprocessing* dan ekstraksi fitur.

3.4 Pengenalan Ekspresi Wajah dengan *Deep Learning*

Pada penelitian ini, *deep learning* adalah kunci utama dari sistem. Oleh karena itu perlu diadakan penelitian yang lebih mendetail agar mendapat model *deep learning* dengan performa baik. Untuk melatih model *deep learning* diperlukan komputer dengan kekuatan GPU yang besar agar lebih mudah dalam tahap eksperimen dan *prototyping*.

3.4.1 Pembuatan Dataset

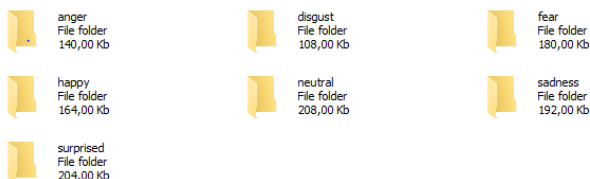
Faktor terbesar dari kualitas model *deep learning* adalah dataset. Semakin representatif dataset yang digunakan, semakin baik performa model. Dalam penelitian ini, pembuatan dataset menjadi fokus utama. Dataset yang digunakan adalah gabungan dari JAFFE [18], CK+ [16], dan sebagian gambar pada database AffectNet [17]. Penggabungan dataset dibutuhkan untuk pelatihan algoritma *deep learning* yang akan diimplementasikan ke sistem pada penelitian ini.

Gambar pada dataset JAFFE dan AffectNet sudah di pisahkan sesuai jenis emosi sebanyak 7 macam yaitu sedih, senang, takut, jijik, marah, terkejut, dan netral dengan cara menyimpan gambar pada *folder* yang

judulnya sesuai dengan ekspresi wajah. Sedangkan pada dataset CK+ masih berbentuk gambar yang dicuplik dari sebuah video secara sekuensial. Oleh karena itu gambar perlu dipilih secara manual dan disortir sesuai jenis emosinya. Total gambar yang ada pada dataset gabungan kurang lebih 40.000 gambar.

Langkah selanjutnya adalah memisahkan gambar wajah dengan latar belakangnya. Hal ini bertujuan untuk agar saat proses *learning* model akan terfokus mempelajari fitur pada wajah. Untuk membantu memisahkan wajah dari latar belakangnya, dibuatlah *python script* dengan bantuan *library* OpenCV dan Pillow (PIL). *Script* ini dapat mendeteksi wajah yang ada pada gambar dan membuat kotak pembatas disekitar wajah. Pemotongan wajah akan sesuai kotak pembatas yang dibuat sebelumnya. Gambar yang sudah terpotong akan dipindah ke folder yang sudah disediakan.

Untuk memastikan kualitas dataset, dilakukan pengecekan gambar satu persatu. Gambar dengan ekspresi wajah kurang sesuai dengan judul *folder* letak penyimpanannya akan dipindahkan ke *folder* yang sesuai. Bila ekspresi wajahnya ambigu, gambar akan dihapus. Pada tahap ini dibutuhkan persepsi manusia dalam mengklasifikasikan ekspresi wajah.

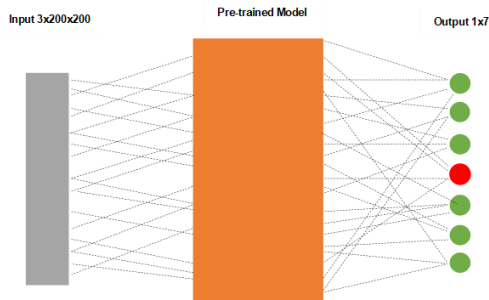


Gambar 3.4 Struktur Penyimpanan Dataset.

3.4.2 Melatih Model *Deep Learning*

Pada bagian ini akan dijelaskan proses dalam melatih model deep learning. Dalam penelitian ini akan dilakukan perbandingan performa model dengan variabel *learning rate*, *decay*, dan *epoch*. Data penting yang akan diambil adalah akurasi dan kecepatan prediksi. Arsitektur CNN yang dipakai adalah MobileNetV2, Xception, dan VGG16 dengan *pre-*

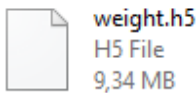
trained weights. Hal ini dilakukan untuk membandingkan model sehingga dapat mengetahui model *deep learning* yang paling sesuai dalam aplikasi penelitian ini. Output dari arsitektur tersebut mempunyai banyak *classes*, oleh karena itu harus dimodifikasi menjadi 7 *classes* terlebih dahulu. Masukan dari model *deep learning* ini adalah gambar berwarna dengan resolusi 200x200. Gambar arsitektur *deep learning* yang telah dimodifikasi bisa dilihat pada gambar 3.3. Output dari training adalah model *weights* dengan format h5 dan json. *Weights* tersebut ukurannya akan diperkecil dan dijalankan pada rangka kerja TensorFlow lite sehingga model dapat dijalankan pada perangkat yang kekuatan komputasinya rendah seperti raspberry pi. Untuk mempermudah pengerjaan pada python, Anaconda digunakan sebagai manajemen paket sehingga lebih mudah dalam menginstal *dependencies* dari *script* latih. Spyder digunakan dalam membuat program latih pada penelitian ini. Spyder mempunyai fitur untuk melihat variabel yang tersimpan pada eksekusi program sehingga dapat mempermudah dalam *debugging*.



Gambar 3.5 Ilustrasi dari Modifikasi Arsitektur Deep Learning.

Layer (type)	Output Shape	Param #
image_input (InputLayer)	(None, 200, 200, 3)	0
mobilenetv2_1.00_224 (Model)	multiple	2257984
flatten (Flatten)	(None, 46080)	0
dropout_1 (Dropout)	(None, 46080)	0
predictis (Dense)	(None, 3)	138243
Total params: 2,396,227		
Trainable params: 2,362,115		
Non-trainable params: 34,112		

Gambar 3.6 Spesifikasi dari MobileNetV2 yang Dimodifikasi pada Keras.



weight.h5
H5 File
9,34 MB



weight
JSON File
79,1 KB

Gambar 3.7 Output dari Pelatihan Model Deep Learning.

```
#Configure Learning Parameter
model.compile(loss='categorical_crossentropy',
              optimizer=optimizers.Adam(lr=0.0001, beta_1=0.9,
                                         beta_2=0.999, epsilon=1e-08,
                                         decay=0.0),
              metrics=['accuracy'])
```

Gambar 3.8 Program Untuk Mengatur Parameter Learning.

3.5 Informasi Suara

Hasil prediksi akan disajikan dalam bentuk file audio untuk tiap ekspresi wajah akan ditugaskan untuk file audio yang berbeda yang dibuat dengan pola sederhana 3 nada yang berbeda kita menyebutnya

audio_A, audio_B, dan audio_C. Daripada menggunakan nada yang berbeda untuk tiap ekspresi wajah, menggunakan pola sederhana dengan nada yang lebih sedikit terasa lebih mudah untuk diingat.

Tabel 3.1. Informasi Audio

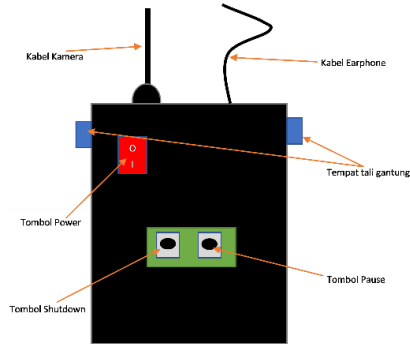
Ekspresi	Representasi Suara	Nama file
Marah	Mainkan audio_A tiga kali	marah.mp3
Jijik	Mainkan audio_A dua kali	jijik.mp3
Takut	Mainkan audio_A	takut.mp3
Netral	Mainkan audio_B	netral.mp3
Senyum	Mainkan audio_C	senyum.mp3
Terkejut	Mainkan audio_C dua kali	terkejut.mp3
Sedih	Mainkan audio_C tiga kali	sedih.mp3

3.6 Desain Perangkat

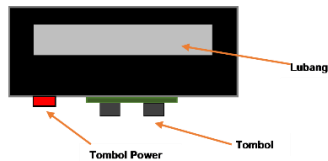
Perangkat keras pada penelitian ini desain elektronik dan rangka perangkat. Pada subbab ini akan dijelaskan rancangan awal dari perangkat ini.

3.6.1 Rancangan Kemasan Perangkat

Wadah perangkat akan sangat diperhatikan pada penelitian ini. Desain wadah harus dibuat sederhana, ringan, dan kecil untuk, memudahkan pengguna. Rancangan awal wadah berbentuk persegi yang dikelungkan pada leher. Bahan untuk membuat persegi tersebut adalah akrilik karena ringan dan mudah dibentuk. Pada penelitian ini Corel Draw digunakan untuk mendesain potongan akrilik. Agar dapat dipakai secara natural, webcam akan dimuat pada sebuah topi. Perekat velcro akan di letakkan pada bawah lidah topi dan salah satu sisi dari webcam. Perekat velcro dikarenakan agar mudah untuk dilepas dan diposisikan ulang sesuai dengan pengguna. Kabel dari webcam akan dikaitkan pada kain pengait yang akan dibuat disekitar topi.



Gambar 3.9 Rancangan perangkat tampak depan.

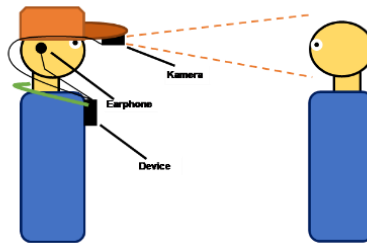


Gambar 3.10 Rancangan perangkat tampak atas.

3.6.2 Skenario Pemakaian

Perangkat berbentuk kotak berukuran kecil yang dikalungkan ke leher agar mudah dipakai saat beraktivitas. Topi digunakan untuk memuat kamera dan *earphone* agar penempatannya tidak mengganggu pengguna. Kamera diletakkan sejajar dengan pandangan mata sehingga citra yang ditangkap kamera sama dengan citra pada mata normal. Citra yang ditangkap digunakan sebagai *input* utama dari sistem ini sehingga penempatannya sangat diperhatikan. *Earphone* digunakan sebagai penyampaian informasi yang didapat oleh sistem. Informasi tersebut berupa nada unik agar penyandang tunanetra dapat membedakan emosi satu dengan yang lainnya. Pada kotak perangkat disediakan tombol jeda

dan *shutdown*. Dengan adanya tombol, pengguna dapat mengatur tempo penggunaan sehingga dapat menghemat baterai.



Gambar 3.11 Ilustrasi skenario pemakaian perangkat.



Gambar 3.12 Contoh pemakaian perangkat

3.7 Pengujian Perangkat

Variabel uji pada penelitian ini adalah pengguna dan kondisi lingkungan uji. Perangkat akan diujikan pada pengguna tunanetra dan pengelihat normal pada ruangan *indoor* maupun *outdoor*. Pengujian ini diharapkan dapat menemukan korelasi dari kategori pengguna alat dengan kemampuan menerima informasi suara. Skenario pengujian akan diperhatikan agar dapat menganalisa data hasil pengujian secara komprehensif.

Subjek pengujian adalah partisipan dengan pengelihan normal dan tunanetra yang diundang sebelumnya. Partisipan dengan pengelihan normal akan berperan sebagai peraga eksperesi wajah dengan dengan sebagian juga merangkap sebagai pengguna perangkat. Partisipan tunanetra hanya berperan sebagai pengguna perangkat.

Sebelum memulai pengujian, partisipan akan ditunjukan informasi suara pada setiap ekspresi dan gambar ekspresi wajah yang ada pada dataset sebagai refrensi dalam mempragakan tiap ekspresi wajah. Ekspresor akan duduk sejauh 1 meter di depan pengguna perangkat. Pengguna perangkat akan menyebutkan informasi ekspresi wajah yang terdeteksi melalui earphone. Persentase keberhasilan dan kesalahan pengguna perangkat akan dicatat. Data yang telah diambil dari setiap pengguna perangkat akan dijadikan satu dan dipresentasikan dengan *confusion matrix*.



Gambar 3.13 Contoh lokasi pengujian *outdoor*.



Gambar 3.14 Contoh lokasi pengujian *indoor*.

BAB 4

HASIL PENGUJIAN

4.1 Kecepatan Prediksi Model *Deep Learning*

Sistem yang diajukan pada penelitian ini menggunakan CNN. CNN dikenal memerlukan kekuatan komputasi yang cukup besar sehingga dapat memperlambat kerja dari suatu sistem. Oleh karena itu, perlu dilakukan percobaan untuk menemukan model yang sesuai dengan aplikasi pada penelitian ini. Untuk memilih model yang sesuai, dilakukan uji kecepatan dari 3 arsitektur *CNN* yaitu MobileNet V2, VGG16, dan Xception. Arsitektur tersebut dilatih pada sebuah dataset tanpa melakukan *tunning* pada parameter *learning*. Pengujian dilakukan dengan memprediksi 50 gambar dengan memperhatikan kecepatan dan akurasi prediksi pada komputer uji dan Raspberry Pi 3.

Tabel 4.1 Hasil pengujian pada komputer uji.

Arsitektur	Kecepatan (detik / prediksi)	Akurasi
VGG16	0.05123	58 %
Xception	0.03244	62 %
Mobilenet V2	0.01982	60 %

Tabel 4.2 Hasil pengujian pada Raspberry Pi 3.

Arsitektur	Kecepatan (detik / prediksi)	Akurasi
VGG16	9.132	53 %
Xception	11.983	56 %
Mobilenet V2	1.2783	55 %

Pada tabel 4.1 perbedaan akurasi dan kecepatan tidak terlalu signifikan. Bila kecepatan model kurang dari 0.1 detik, maka model tersebut sudah sangat baik untuk memprediksi gambar secara kontinu. Tidak ada perbedaan akurasi secara signifikan dari 3 arsitektur tersebut. Pada tabel 4.2 terdapat perbedaan yang sangat signifikan pada kecepatan. Berdasarkan pengamatan, Mobilenet V2 sangat unggul pada kecepatan prediksi meskipun belum bisa dibilang mempunyai kecepatan yang baik.

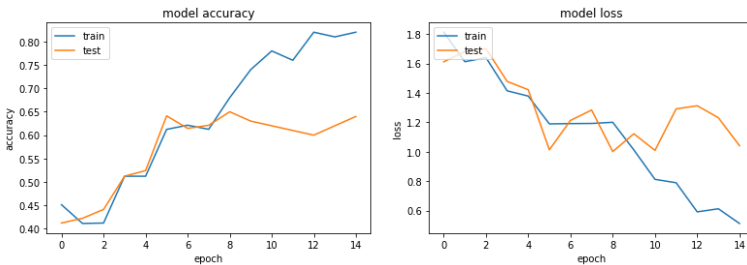
Akan tetapi, dalam aplikasi penelitian ini kecepatan tersebut sudah memenuhi. Dalam aspek akurasi, ke tiga model mempunyai kecepatan yang relatif sama. Dari ketiga arsitektur, MobileNet V2 yang paling sesuai untuk digunakan pada penelitian ini.

4.2 Hyperparameter Tuning

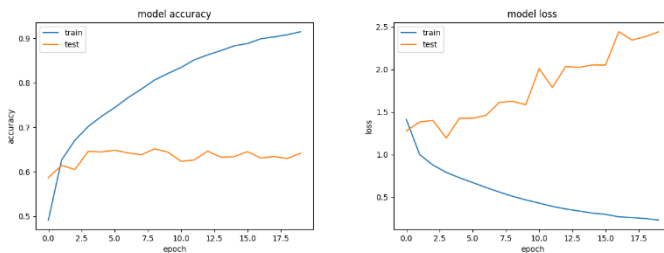
Pada penelitian ini akan diadakan pengaturan parameter *learning*. Parameter yang akan diperhatikan adalah *learning rate* dan *momentum*. Pada pengujian, model MobileNet V2 akan dilatih menggunakan dataset yang sudah dikumpulkan. Pertama, model akan dilatih beberapa kali dengan mengubah parameter *momentum* untuk mendapatkan acuan nilai optimal. Hal yang sama akan dilakukan untuk mencari nilai *learning rate* yang optimal. Setelah menemukan nilai acuan yang optimal, dilakukan *learning* model dengan nilai acuan yang telah ditemukan sebelumnya.

4.2.1 Tuning Momentum

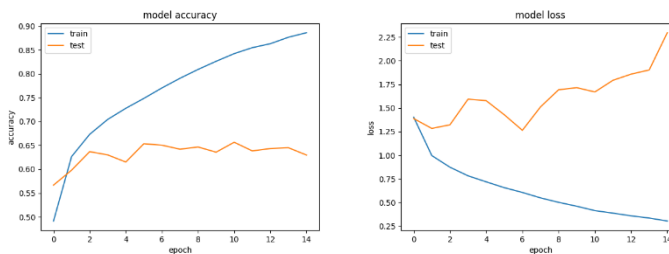
Momentum dapat membantu untuk mengetahui arah perubahan parameter model yang mengacu pada hasil langkah sebelumnya dan juga dapat mencegah osilasi pada *gradient descent*. Pada bagian ini, akan dilatih model *deep learning* dengan mencoba beberapa nilai *momentum*.



Gambar 4.1 Riwayat learning dengan nilai momentum 0,5.



Gambar 4.2 Riwayat learning dengan nilai momentum 0,7.



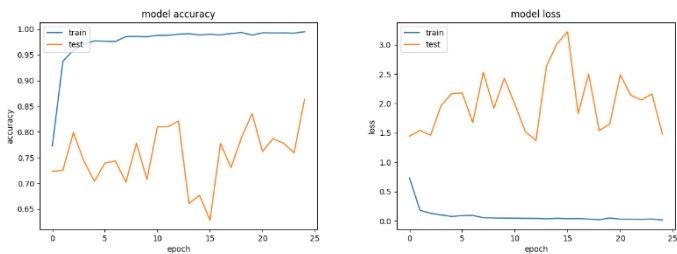
Gambar 4.3 Riwayat learning dengan nilai momentum 0,9.

Berdasarkan gambar di atas, bisa disimpulkan bahwa dengan mengubah nilai *momentum* dapat mempengaruhi sesi *learning* model secara signifikan. Pada nilai 0,9 dan 0,7, akurasi pada data latih meningkat hingga mencapai 0,9 pada *epoch* 15 dan perubahan pada data uji kurang signifikan. *Loss* dari data latih pada nilai tersebut (0,9 dan 0,5) meningkat seiring berjalannya sesi *learning*. Hal ini menyebabkan tidak adanya peningkatan akurasi pada data uji. Dari kecenderungan tersebut bisa disimpulkan bahwa nilai 0,9 dan 0,7 kurang cocok dalam aplikasi penelitian ini atau bisa disebut *overfitting*. Pada nilai *momentum* 0,5 akurasi data latih meningkat secara bertahap akan tetapi pada epoch yang sama akurasi sesi latih dari nilai *momentum* 0,5 tidak setinggi dengan nilai yang lain (0,9 dan 0,7). Akan tetapi, akurasi pada data uji cenderung meningkat. *Loss* pada data uji berkurang seiring berjalannya sesi *learning*.

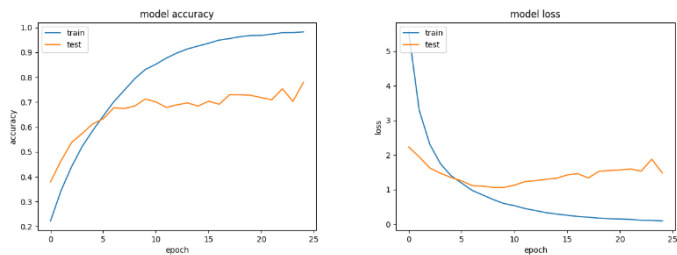
Dari kecenderungan tersebut, bisa disimpulkan bahwa nilai *momentum* yang cocok untuk aplikasi ini kurang lebih 0,5.

4.2.2 Tuning Learning Rate

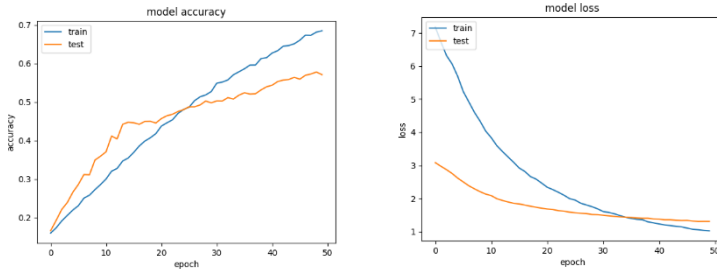
Setelah memilih nilai *momentum*, akan dilakukan pengaturan parameter *learning rate*. *Learning rate* menentukan seberapa cepat model memperbarui parameter model. Dengan menemukan nilai *learning rate* yang sesuai, model dapat mempelajari fitur input dengan baik dan cepat. Pada penelitian ini, akan merekam riwayat sesi latih dengan nilai learning rate yang berbeda.



Gambar 4.4 Riwayat learning dengan nilai learning rate 0,001.



Gambar 4.5 Riwayat learning dengan nilai learning rate 0,0001.



Gambar 4.6 Riwayat learning dengan nilai learning rate 0,00001.

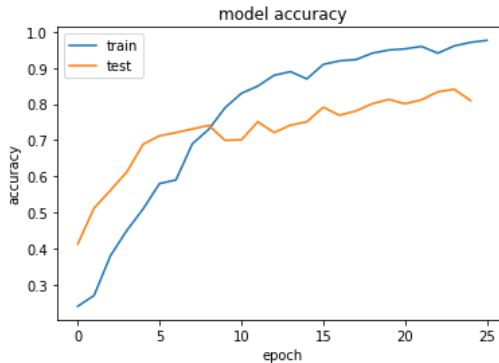
Berdasarkan gambar diatas, perubahan nilai *learning rate* dapat mempengaruhi sesi *learning*. Pada nilai 0,001 model mengalami kesulitan dalam mengenali data tes. Bisa dilihat pada riwayat training nilai 0,001 mempunyai loss yang sangat tinggi meskipun mempunyai akurasi latih yang tinggi dan loss latih yang rendah. Dari karakteristik tersebut, bisa disimpulkan bahwa model mengalami *overfitting* bila menggunakan nilai learning rate sebesar 0,001.

Pada nilai 0,0001 dan 0,00001 mempunyai kecenderungan yang sama yaitu akurasi data latih dan tes menunjukkan peningkatan. Seiring berjalannya sesi learning, loss data tes dan latih mengalami penurunan. Akan tetapi pada nilai 0,00001 membutuhkan sesi learning yang lebih lama untuk menyamai kemampuan model pada nilai 0,0001. Bila diamati lebih detail, akurasi data latih menunjukkan stagnansi pada beberapa *epoch* terakhir. Sehingga, bila ingin menyamai performa model dengan nilai *learning rate* 0,0001 akan memakan waktu yang sangat lama. Oleh karena itu, nilai *learning rate* yang sesuai dengan penelitian ini adalah kurang lebih 0,0001.

4.2.3 Model Akhir

Setelah mendapatkan acuan nilai variabel *momentum* dan *learning rate* akan dilakukan sesi *learning* ulang. Model dari sesi learning ini akan menjadi model final yang akan diimplementasikan ke sistem bila mencapai batas minimum akurasi yang diinginkan yaitu 70% dari data

tes. Harapan dari *hyperparameter tuning* adalah mendapatkan model yang akurasi lebih dari batas minimum. Gambar 4.7 adalah riwayat sesi training yang sudah dilakukan *hyperparameter tuning* dengan nilai variabel momentum dan learning rate masing masing 0,55 dan 0,0004. Akurasi data latih hamper mencapai 0.99.



Gambar 4.7 Riwayat *learning* setelah melakukan *hyperparameter tuning*.

4.3 Pengujian Perangkat Pada Pengguna Dengan Pengelihan Normal

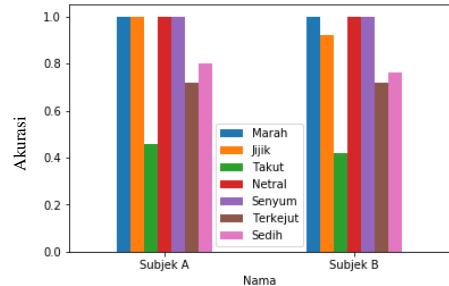
Pada bagian ini alat akan diuji ke pengguna dengan pengelihan normal pada kondisi lingkungan yang berbeda (ruangan terbuka dan ruangan tertutup).

4.3.1 Pengujian Di Dalam Ruangan

Pada bagian ini sistem akan diuji di dalam ruangan. Tambahan pencahayaan hanya 1 lampu meja. Dalam pengujian ini, diundang 10 partisipan sebagai ekspresor dan dua orang merangkap sebagai pengguna perangkat. Partisipan yang bertindak sebagai ekspresor akan memperagakan 7 ekspresi wajah sebanyak 5 kali. Pengguna perangkat akan menebak ekspresi yang dilakukan oleh 5 orang ekspresor.



Gambar 4.8 Pengujian sistem rekognisi ekspresi wajah dalam ruangan.



Gambar 4.9 Diagram persentase keberhasilan prediksi pada pengujian dalam ruangan.

		Predicted Class						
		Anger	Disgust	Fear	Neutral	Happy	Surprised	Sadness
Actual Class	Anger	1	-	-	-	-	-	-
	Disgust	0.04	0.96	-	-	-	-	-
	Fear	-	-	0.54	0.1	0.26	0.1	-
	Neutral	-	-	-	1	-	-	-
	Happy	-	-	-	-	1	-	-
	Surprised	-	-	0.1	0.18	-	0.72	-
	Sadness	0.12	0.06	-	0.08	-	-	0.74

Gambar 4.10 Confusion matrix dari hasil pengujian sistem rekognisi ekspresi wajah dalam ruangan.

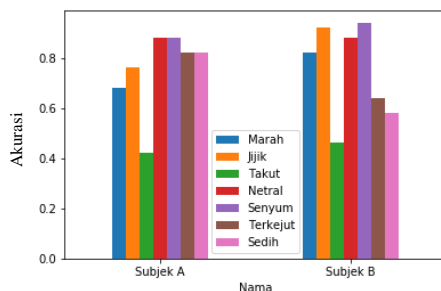
Harapan dari pencahayaan yang bagus adalah agar mendapat akurasi prediksi yang baik. Pada Gambar 4.10, persentasi keberhasilan kedua subjek tidak jauh berbeda. Perbedaan yang paling terlihat ada pada ekspresi jijik, tetapi perbedaan tersebut tidak terlalu berpengaruh ke performa perangkat secara menyeluruh. Dilihat pada Gambar 4.11, ekspresi marah, netral dan senyum tidak pernah mengalami kesalahan saat prediksi. Ekspresi takut memiliki kesalahan prediksi terbanyak pada pengujian ini. Kesalahan prediksi pada setiap ekspresi dicatat dengan harapan menemukan suatu pola kesalahan dan penyebabnya. Pada kondisi dalam ruangan, sistem mempunyai akurasi prediksi total 85,14% dan bisa dikatakan sistem berjalan dengan baik.

4.3.2 Pengujian Di Luar Ruangan

Pada bagian ini sistem akan diuji di luar ruangan untuk mensimulasikan keadaan lingkungan dengan pencahaya yang tidak konsisten. Dalam pengujian ini, diundang 5 partisipan sebagai ekspresor dan dua orang merangkap sebagai pengguna perangkat. Partisipan yang bertindak sebagai ekspresor akan memperagakan 7 ekspresi wajah sebanyak 5 kali. Pengguna perangkat akan menebak ekspresi yang dilakukan oleh 4 orang ekspresor.



Gambar 4.11 Pengujian sistem rekognisi ekspresi wajah luar ruangan.



Gambar 4.12 Diagram persentase keberhasilan prediksi pada pengujian luar ruangan.

		Predicted Class						
		Anger	Disgust	Fear	Neutral	Happy	Surprised	Sadness
Actual Class	Anger	0.80	0.20	-	-	-	-	-
	Disgust	0.16	0.84	-	-	-	-	-
	Fear	0.04	-	0.48	-	0.2	0.12	0.16
	Neutral	-	-	0.04	0.88	-	-	0.08
	Happy	-	-	0.04	-	0.92	0.04	-
	Surprised	0.04	0.08	0.04	-	0.08	0.76	-
	Sadness	0.04	0.08	-	0.24	-	-	0.64

Gambar 4.13 Confusion matrix dari hasil pengujian sistem rekognisi ekspresi wajah luar ruangan.

Pada Gambar 4.12, tingkat keberhasilan prediksi dari kedua pengguna perangkat ada perbedaan di beberapa kategori ekspresi. Akan tetapi kedua subjek memiliki trend keberhasilan yang serupa. Dilihat pada Gambar 4.13, ekspresi senyum memiliki tingkat kesalahan prediksi paling rendah. Ekspresi takut memiliki kesalahan prediksi terbanyak pada pengujian ini. Untuk ekspresi lainnya, terdapat kesalahan prediksi tetapi masih pada batas wajar. Pada kondisi luar ruangan, sistem mempunyai akurasi prediksi total 76% dan bisa dikatakan sistem berjalan dengan baik.

4.4 Pengujian Perangkat Pada Pengguna Tunanetra

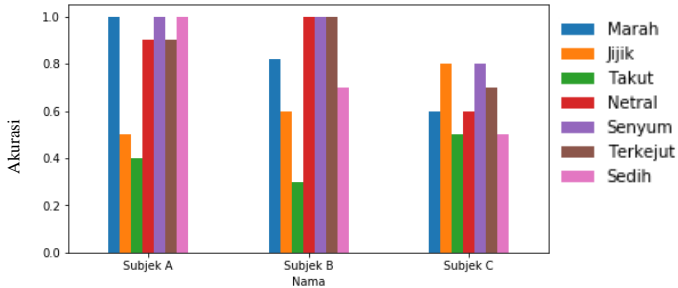
Pada bagian ini alat akan diuji ke pengguna dengan pengelihatn normal pada kondisi lingkungan yang berbeda (ruangan terbuka dan ruangan tertutup).

4.4.1 Pengujian Di Dalam Ruangan

Pada pengujian ini sistem rekognisi ekspresi wajah sudah diimplementasikan secara lengkap menjadi sebuah perangkat portabel. Perangkat akan diuji akurasi prediksinya. Pengujian akan dilakukan oleh 3 penyandang tunanetra sebagai pengguna perangkat dan 4 orang dengan pengelihatn normal sebagai objek untuk direkognisi ekspresi wajahnya. Partisipan akan memperagakan 7 ekspresi wajah sebanyak 5 kali kepada 3 pengguna perangkat. Perangkat akan dipakai di ruangan tertutup tanpa tambahan lampu.



Gambar 4.14 Pengujian perangkat pada penyandang tunanetra di ruangan tertutup.



Gambar 4.15 Diagram persentase keberhasilan prediksi pengguna tunanetra pada ruangan tertutup.

		Predicted Class						
Actual Class		Anger	Disgust	Fear	Neutral	Happy	Surprised	Sadness
	Anger	1	-	-	-	-	-	-
	Disgust	0.44	0.56	-	-	-	2	-
	Fear	-	-	0.32	0.16	0.32	0.2	-
	Neutral	-	-	-	0.96	0.04	-	-
	Happy	-	-	-	-	1	-	-
	Surprised	-	0.04	-	-	0.04	0.92	-
	Sadness	-	-	-	0.16	-	-	0.84

Gambar 4.16 Confusion matrix dari hasil pengujian perangkat ruangan tertutup.

Pada gambar 4.15, subjek A dan B memiliki tingkat keberhasilan yang relatif sama. Akan tetapi adanya perbedaan yang signifikan dengan subjek C. Setelah diadakan tanya jawab pada subjek C, dijelaskan bahwa subjek kurang menguasai pada pengenalan informasi melalui suara. Namun, subjek C merasa lebih nyaman setelah pengambilan data pada ekspresor kedua. Dilihat dari Gambar 4.16, pengguna berhasil menebak ekspresi wajah dengan benar sebesar 80% dengan kondisi ruangan yang natural tanpa adanya tambahan pencahayaan. Ekspresi takut menjadi ekspresi wajah yang memiliki kesalahan yang paling besar. Pada ekspresi jijik, eror dari jawaban selalu berada di ekspresi marah. Hal ini diduga dikarenakan ekspresi marah dan jijik memiliki fitur wajah yang mirip atau pengguna kurang tepat dalam mencerna informasi suara dikarenakan

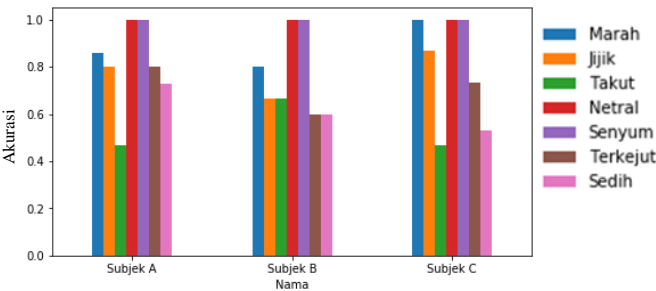
ekspresi marah mempunya pola suara yang hampir sama dengan ekspresi jijik (dilihat Tabel 3.1).

4.4.2 Pengujian Di Luar Ruangan

Pengujian akan dilakukan oleh 3 penyandang tunanetra sebagai pengguna perangkat dan 3 orang dengan pengelihatan normal sebagai objek untuk direkognisi ekspresi wajahnya. Partisipan akan memperagakan 7 ekspresi wajah sebanyak 5 kali kepada 3 pengguna perangkat. Perangkat akan dipakai di ruangan terbuka tanpa tambahan lampu.



Gambar 4.17 Pengujian perangkat pada penyandang tunanetra di ruangan terbuka.



Gambar 4.18 Diagram persentase keberhasilan prediksi pengguna tunanetra pada ruangan terbuka.

		Predicted Class						
		Anger	Disgust	Fear	Neutral	Happy	Surprised	Sadness
Actual Class	Anger	0.889	0.112	-	-	-	-	-
	Disgust	0.112	0.778	-	-	0.044	-	0.066
	Fear	0.177	0.112	0.533	-	0.112	-	-
	Neutral	-	-	-	1	-	-	-
	Happy	-	-	-	-	1	-	-
	Surprised	-	0.066	-	0.066	0.1334	0.711	-
	Sadness	-	0.377	-	-	-	-	0.622

Gambar 4.19 Confusion matrix dari hasil pengujian perangkat ruangan tertutup.

Pada gambar 4.18, subjek A, B, C memiliki tren yang hampir sama pada persentase keberhasilan prediksi. Ekspresi takut memiliki persentase keberhasilan terendah. Ekspresi netral dan senyum selalu menduduki peringkat pertama pada persentase keberhasilan prediksi. Dilihat dari Gambar 4.19, pengguna berhasil menebak ekspresi wajah dengan benar sebesar 78% pada ruangan terbuka dengan tanpa adanya tambahan pencahayaan.

Halaman ini sengaja dikosongkan

BAB 5 PENUTUP

5.1 Kesimpulan

Dari hasil penelitian dan analisis, didapatkan kesimpulan sebagai berikut.

1. Informasi ekspresi melalui suara dapat yang dirancang pada penelitian ini memiliki akurasi rata-rata mencapai 80 %.
2. Sistem rekognisi ekspresi wajah dalam penelitian ini performanya menurun hingga 7 % saat dipakai pada ruangan terbuka.
3. Dengan *tuning* parameter *learning* dapat meningkatkan akurasi prediksi model mencapai 30 %.

5.2 Saran

1. Dataset dapat diperkaya dengan gambar berpencahayaan yang pembuatan informasi suara yang paling sesuai dengan pengguna.kontrasnya tidak merata.
2. *Tuning* parameter dapat diperdalam pada pemilihan *optimizer learning* dan *decay*.
3. Sensitifitas prediksi perangkat perlu disesuaikan lagi.
4. Perlu diadakan analisa lebih dalam pada cara menyampaikan informasi ekspresi pada tunanetra.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Buduma, Nikhil, dan Nicholas Locasio, “*Fundamentals of Deep Learning*”, USA, O’Reilly, 2017
- [2] Hung, C.-Y., Chen, W.-C., Lai, P.-T., Lin, C.-H., & Lee, C.-C., “Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database, ” *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017
- [3] Nash, Ryan, dan Keiron O’Shea , "An Introduction to Convolutional Neural Networkw", ResearchGate, Desember, 2015
- [4] Chollet, François, "Xception: Deep Learning with Depthwise Separable Convolutions", *Proc. - 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800 - 1807, 2017
- [5] SH Tsang, " Review: Xception—With Depthwise Separable Convolution, Better Than Inception-v3 (Image Classification)", 2018. [Online]. Available: <https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568>. [Accessed: 22-May-2019]
- [6] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", *Proc. - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510 - 4520, 2018
- [7] Simonyan, Karen, dan Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv:1409.1556, September, 2014
- [8] Szeliski, Richard, "Computer Vision: Algorithm and Applications", Springer, 2010
- [9] Google Brain Team, “TensorFlow”, 2015. [Online]. Available: <https://www.tensorflow.org/>. [Accessed: 21-May-2019].

- [10] Wikipedia, "TensorFlow", 2015. [Online]. Available: <https://en.wikipedia.org/wiki/TensorFlow>. [Accessed: 19-May-2019].
- [11] Google Brain Team, "TensorFlow Lite", 2018. [Online]. Available: <https://www.tensorflow.org/lite/>. [Accessed: 21-May-2019].
- [12] Keras, "Keras Documentation". [Online]. Available: <https://keras.io/>. [Accessed: 18-May-2019].
- [13] Raspberry Pi Foundation, "Raspberry Pi", 2012. [Online]. Available: <https://www.raspberrypi.org>. [Accessed: 18-May-2019].
- [14] OpenCV Developer Team, "About OpenCV," 2017. [Online]. Available: <http://opencv.org/about.html>. [Accessed: 21-May-2019].
- [15] Wikipedia, "Anaconda" 2014. [Online]. Available: [https://en.wikipedia.org/wiki/Anaconda_\(Python_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)). [Accessed: 22-May-2019].
- [16] Lucey, Patrick, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," *Proc. - 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops* , pp. 96-101, 2010
- [17] Mollahosseini, Ali, Behzed Hasani, Mohammad H. Mahoor, "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild, " *IEEE Transactions on Affective Computing*, pp. 99 , 2017
- [18] M. J. Lyons, M. Kamachi and J. Gyoba, "Japanese Female Facial Expressions (JAFPE)," *Database of digital images*, 1997
- [19] Matthew N Dailey, Carrie Joyce, Michael J Lyons, Miyuki Kamachi, Hanae Ishi, Jiro Gyoba, Garrison W Cottrell, "Evidence and a computational explanation of cultural differences in facial expression recognition, " *Emotion*, Vol 10(6), hlm. 874-893, Dec 2010,
- [20] Zhao, Yuhang, Shaomei Wu, Lindsay Reynolds, dan Shiri Azenkot, "A Face Recognition Application for People with Visual Impairments: Understanding Use Beyond the Lab", *Conference on Human Factors in Computing Systems*, April 2018

- [21] Jin, Yongsik, Jonghong Kim, Bumwhi Kim, Rammohan Mallipeddi, dan Minho Lee, “Smart Cane: Face Recognition System for Blind”, Conference on Human-Agent Interaction, Daegu, Oktober, 2015
- [22] Bittner, Marian, Hendrik P. Buimer, Tjerk Kostelijk, Thea M. van der Geest, Abdellatif Nemri, Richard J.A. van Wezel, Yan Zhao, “Conveying Facial Expressions to Blind and Visually Impaired Persons Through a Wearable Vibrotactile Device”, PLoS One, Maret, 2018

LAMPIRAN A

1. Program latihan *deep learning*:

```
from keras.applications.mobilenet_v2 import MobileNetV2
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Input, Flatten, Dense, Dropout
from keras.models import Model
from keras import optimizers
import matplotlib.pyplot as plt

## other
img_width, img_height = 200, 200
nb_train_samples = 2000
nb_validation_samples = 350
top_epochs = 50
fit_epochs = 50
batch_size = 8
nb_classes = 3
nb_epoch = 15

#build CNN

model = MobileNetV2(weights='imagenet', include_top=False)

input = Input(shape=(img_width, img_height, 3), name =
'image_input')

output = model(input)

model.summary()

x = Flatten(name='flatten')(output)
#x = Dense(256, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(3, activation='softmax', name='predictis')(x)
```

```

mobilenetv2_model = Model(inputs=input, outputs=x)

mobilenetv2_model.summary()

#Image preprocessing and image augmentation with keras
mobilenetv2_model.compile(loss='categorical_crossentropy',
                           optimizer=optimizers.Adam(lr=0.0001, beta_1=0.9,
beta_2=0.999, epsilon=1e-08, decay=0.0),
                           metrics=['accuracy']
)

# Setting learning data
train_datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('dataset/train',
                                                target_size = (img_width,
img_height),
                                                batch_size = 16,
                                                class_mode = 'categorical')

test_set = test_datagen.flow_from_directory('dataset/val',
                                            target_size = (img_width, img_height),
                                            batch_size = 16,
                                            class_mode = 'categorical')

y_true_labels = training_set.class_indices

history = mobilenetv2_model.fit_generator(

```

```

        training_set,
        steps_per_epoch=nb_train_samples,
        epochs=nb_epoch,
        validation_data=test_set,
        validation_steps=nb_validation_samples
    )

    print(history.history.keys())
    # summarize history for accuracy
    plt.plot(history.history['acc'])
    plt.plot(history.history['val_acc'])
    plt.title('model accuracy')
    plt.ylabel('accuracy')
    plt.xlabel('epoch')
    plt.legend(['train', 'test'], loc='upper left')
    plt.savefig("acc.png")
    plt.show()

    # summarize history for loss
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['train', 'test'], loc='upper left')
    plt.savefig("loss.png")
    plt.show()

    #Save the model
    # serialize model to JSON
    my_model_json = mobilenetv2_model.to_json()
    with open("mobilenetv2_model.json", "w") as json_file:
        json_file.write(my_model_json)
    # serialize weights to HDF5
    mobilenetv2_model.save_weights("mobilenetv2_model.h5")
    print("Saved model to disk")

```

2. Program rekognisi pada komputer:

```
json_file.close()

loaded_model = model_from_json(loaded_model_json)
loaded_model.load_weights('mobilenetv2_model.h5')
print("Loaded model from disk")

# Create the haar cascade
faceCascade = cv2.CascadeClassifier(cascPath)

cap = cv2.VideoCapture(0)

img = np.zeros((200, 200, 3))
ct = 0
while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()
    ct+=1

    # Our operations on the frame come here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Detect faces in the image
```

```

faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize=(150, 150)
    #flags = cv2.CV_HAAR_SCALE_IMAGE
)

ano = "
for (x, y, w, h) in faces:

    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    img = crop_center(gray, x, y, w, h)
    img = resize(img,(200,200, 3))
    img = np.expand_dims(img,axis=0)
    if(np.max(img)>1):
        img = img/255.0

    res = loaded_model.predict(img)

    classes = np.argmax(res,axis=1)

    print (classes)

```



```

# Display the resulting frame
cv2.imshow('frame',frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()

3.    Program rekognisi pada raspberry pi:
import cv2
import sys
import tensorflow as tf
import numpy as np
import RPi.GPIO as GPIO
from skimage.transform import resize
import statistics
import os
import time

def crop_center(img, x, y, w, h):
    return img[y:y+h,x:x+w]

def preprocess_img(raw):

```

```

img = resize(raw,(200,200, 3))
img = np.expand_dims(img,axis=0)
if(np.max(img)>1):
    img = img/255.0
return img

```

```

def brain(raw):
    ano = "
    img = crop_center(raw, x, y , w , h)
    img = preprocess_img(img)
    f.set_tensor(i['index'], img.astype(np.float32))
    f.invoke()
    res = f.get_tensor(o['index'])
    classes = np.argmax(res,axis=1)
    return classes[0]

```

```

os.system("mpg123 assets/audio/" + '1.mp3')

```

```

print('Loading ..')

```

```

f = tf.contrib.lite.Interpreter("assets/model/model_optimized.tflite")
f.allocate_tensors()
i = f.get_input_details()[0]

```

```

o = f.get_output_details()[0]

print('Load Success')

cascPath = "haarcascade_frontalface_default.xml"

faceCascade = cv2.CascadeClassifier(cascPath)

GPIO.setwarnings(False) # Ignore warning for now
GPIO.setmode(GPIO.BOARD) # Use physical pin numbering
GPIO.setup(10, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(12, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

cap = cv2.VideoCapture(0)
res = []
bef = 'anger'
img = np.zeros((200, 200, 3))
ct = 0
last_state = 1
activate = True
countdown = 0

while(True):

```

```

state = GPIO.input(10)

if ((not last_state) and state):
    if activate == 1:
        os.system("mpg123 assets/audio/" + 'pause.mp3')
    else :
        os.system("mpg123 assets/audio/" + 'resume.mp3')
    activate = not activate
    ct = 0
    res = []

if GPIO.input(12) == 1:
    countdown+=1
else :
    countdown = 0

if countdown > 50:
    os.system("mpg123 assets/audio/" + '2.mp3')
    time.sleep(2)
    os.system("sudo shutdown -h now")
last_state = state
time.sleep(0.05)

```

```

if activate == True:
    ret, frame = cap.read()
    ct+=1

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(150, 150)
    )
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
        if ct > 3:
            ai = brain(gray)
            res.append(ai)
            if len(res) >= 5:
                try:
                    mode = statistics.mode(res)
                except ValueError:
                    mode = bef

```

```

if bef != mode:

    print(mode)

    if mode == 0:

        os.system("mpg123 assets/audio/" + 'marah.mp3')

    elif mode == 1:

        os.system("mpg123 assets/audio/" + 'jijik.mp3')

    elif mode == 2:

        os.system("mpg123 assets/audio/" + 'takut.mp3')

    elif mode == 3:

        os.system("mpg123 assets/audio/" + 'senyum.mp3')

    elif mode == 4:

        os.system("mpg123 assets/audio/" + 'neutral.mp3')

    elif mode == 5:

        os.system("mpg123 assets/audio/" + 'sedih.mp3')

    else :

        os.system("mpg123 assets/audio/" + 'terkejut.mp3')

    bef = mode

    res = []

else:

    res.pop(0)

ct = 0

cap.release()

cv2.destroyAllWindows()

```

LAMPIRAN B

1. Contoh wajah yang ditangkap oleh perangkat dalam ruangan tertutup.



2. Dokumentasi pengujian perangkat ke pengguna dengan pengelihan normal pada ruangan tertutup.



2. Dokumentasi pengujian perangkat ke pengguna penglihatan normal pada ruangan terbuka.



3. Dokumentasi pengujian perangkat lunak pada tunanetra.



3. Dokumentasi pengujian perangkat pada tunanetra di dalam ruangan.



Halaman ini sengaja dikosongkan

BIODATA PENULIS



Hasby Fahrudin lahir dan dibesarkan di Surabaya sejak 22 tahun lalu. Penulis menyelesaikan pendidikan dasar di SDIT Ghilmani Surabaya pada tahun 2009. Setelah itu, penulis melanjutkan pendidikan di SMPN 6 Surabaya dan SMAN 2 Surabaya. Setelah lulus jenjang SMA, penulis melanjutkan ke jenjang perguruan tinggi di Institut Teknologi Sepuluh Nopember melalui jalur SBMPTN. Pada masa perkuliahan penulis banyak menghabiskan waktu di workshop elektro. Penulis menjadi asisten laboratorium elektronika dasar sejak semester 6. Pernah menjadi peserta GKS ASEAN SCIENS di Korea Selatan. Suka makanan pedas dan menonton film bergenre *thriller*. Penulis bercita-cita bekerja di perusahaan berbasis di Silicon Valey.